Eksamen D

I fagene

Programmering og Teknik

2023, juni, kl. 9.00 – 13.00 (14.30)

Alle skriftlige materialer, pc'er, bærbare computere og internet ressourcer er tilladt til eksamen.

Mobiltelefoner og kommunikation med andre personer (inkl. chatbots), bortset fra kommunikation med eksaminator og censor, er forbudt. Dog er det tilladt at bruge GitHub Copilot

Du må ikke gemme dine løsninger på eksterne netværksdrev/værter som GitHub, Facebook o. lign. Overtrædelse af dette vil medføre bortvisning fra eksamen og passende sanktioner vil senere blive tildelt både til afsender/uploader og modtager.

Ved eksamens afslutning skal du lægge din løsning op i Wiseflow.

Eksamen varer 4 timer og efterfølges af 1 times evaluering. I disse tidsrum kan du kun forlade eksamenslokalet med henblik på toilet besøg. Rygning er ikke tilladt.

Ud over disse opgaver, kan du kan blive bedt at besvare nogle ekstra spørgsmål omkring dine svar og eventuelle andre valg.

Kontroller at dette sæt indeholder sider inklusive forsiden.

# Introduktion

I denne opgave skal du lave et simpelt proxysystem. Du skal arbejde med REST API, Python, JavaScript og Selenium

Du skal lave et system der kan holde registrere et vandforbrug, WaterFlow.

Der skal registreres følgende:

* Id, et tal
* Name, en tekst-streng
* Volume, et tal

Opgave 1 REST API
Du skal lavet et REST API, der kan WaterFlow objekter.

1. Lav et nyt projekt af typen ”ASP.NET Core Web API”.
Husk at vælge API og ikke Web Application, når du laver projektet.
Sørg for at “Configure for HTTPS” er unchecked (du skal ikke bruge HTTPS)
2. Lav en model klasse, WaterFlow med de 3 properties (Id, Name og Volume)

***Bemærk:*** *du skal ikke lave validate metoder på din model klasse*

1. Opret en repository klasse WaterFlowsRepository. Repository klassen skal bruge en liste til opbevare WaterFlow objekter.
Du skal indsætte mindst 3 forskellige WaterFlows objekter i listen. Id skal tildeles af repository klassen.
2. Repository klassen skal have følgende metoder
- GetAll(), returnerer alle WaterFlow objekterne
- GetById(id), returnerer det WaterFlow objekt der har det angivne id
- Add(WaterFlow), tilføjer et nyt WaterFlow objekt til listen
3. Lav en controller klasse, WaterFlowsController.
Metoderne i controller klassen skal kalde metoderne i repository klassen.
4. Afprøv dit REST API med Swagger

# Opgave 2 Python UDP Broadcast

Du skal lave en ny Python fil WaterFlowBroadcast. Denne skal kunne simulere vandmåleren der sender aktuelt vandforbrug

1. Spørg brugeren efter vandmålerens navn (name) (input)
2. Lav en løkke der aldrig stopper
3. Inde i løkken skal du generere en tilfældig værdi mellem 1 og 10 og gemme den som volume
4. Opret et objekt med name og volume
5. Serialisere dit objekt om til JSON
6. Send dit objekt som broadcast på port 13472
7. Løkken skal ”sove” i 2 sekunder

Vis i SocketTest at din broadcast sender data.

Hints:

* Random – import random og benyt random.randint(fra, til)
* Json– import json og benyt json.dumps til at serialisere dit objekt om til JSON
* Tid – import time og benyt time.sleep(sek)

# Opgave 3 Python proxy

Du skal lave endnu en Python fil WaterFlowProxy. Denne skal fungere som proxy imellem din Broadcast og dit REST API

1. Lyt efter UDP pakker på port 13472
2. Hver gang en pakke modtages, skal du deserialiser den om til et Python objekt
3. Så skal du videresende pakken til dit REST API

Hints:

* Json– import json og benyt json.loads til at deserializere dit JSON om til et objekt
* Requests – import requests og benyt requests.post(api\_url, json=data) til at sende data til dit REST API

# Opgave 4 CORS og Azure

1. Du skal tilføje CORS til dit REST API.
2. Du skal publicere din REST API til Azure. Sørg for at Swagger virker i Azure.

# Opgave 5 Web-Applikation

Du skal lave en web-applikation med HTML, JavaScript, Vue.js, Axios og Bootstrap.

Din web-applikation skal kalde metoderne fra dit REST API.

Brugeren af web-applikationen skal kunne

1. Se en liste med alle WaterFlow objekterne
2. Vise et WaterFlow objekt med et angivet Id
3. Tilføje et nyt WaterFlow objekt

Brug Bootstrap til at *style* din web-applikation

# Opgave 6 Web-applikation til Azure

Du skal publicere din web-applikation til Azure.

Vis at det virker.

# Opgave 7 Flere feautures

Du skal tilføje en DateTime property til dine WaterFlow objekter, den skal indeholde den aktuelle dato og klokkeslæt. Du bestemmer selv hvor det bliver tilføjet, Python BroadCast, Python Proxy eller REST API. Begrund dit valg.

Opdater din Web-Applikation så Datetime propertien bliver vist, og kan udfyldes ved oprettelse af nyt objekt.